CPE/EE 422/522 SP2004, Lab Assignment 4

4-bit Signed Binary Number (2 s complement) Multiplier

(Undergraduate 100 points -- Graduate 80 points)

The purpose of this laboratory project is to give each student the opportunity to develop a practical logic design using both schematic capture and/or VHDL that will implement a 4-bit multiplier for signed binary number using Booth s algorithm. The design will take two 4-bit singed binary numbers (2 s complement) from Altera UP 1 Educational Trainer's FLEX_SW1 switches and display the product in decimal format as well as the sign on two of the Altera UP 1 Educational Trainer's seven-segment LEDs.

Background

Problem 4.15 (pages 158-159) of the textbook has a detailed description of the Booth s algorithm. You can try this nice on-line Booth s algorithm simulator also: (http://www.ecs.umass.edu/ece/koren/arith/simulator/Booth/)

Pin Assignment

Altera Pin Numbers for FLEX_PB1 push button

FLEX_PB1 push button connects to pin 28 of the EPF10K20 FPGA device

Altera Pin Numbers for Crystal Oscillator

The Altera UP 1 Educational Trainer's board contains a 25.175-MHz crystal oscillator. The output of the oscillator drives a global clock input on the EPF10K20 FPGA device (Pin91).

Altera Pin Numbers for the FLEX_SW1 switches connections

FLEX_SW1 Pin Assignment	
Switch	**EPF10K20 Pin**
FLEX_SW1-1	41
FLEX_SW1-2	40
FLEX_SW1-3	39
FLEX_SW1-4	38
FLEX_SW1-5	36
FLEX_SW1-6	35
FLEX_SW1-7	34
FLEX_SW1-8	33

Note: use SW1- SW4 to set up multiplier, and SW5 — SW8 to set up multiplicand

Altera Pin Numbers for the FLEX DIGIT Segment I/O Connections

FLEX_DIGIT Segment I/O Connections		
Display segment	**Pin for Digit 1**	**Pin for Digit 2**
A	6	17
B	7	18
C	8	19
D	9	20
E	11	21
F	12	23
G	13	24
Decimal Point	14	25

Note: Decimal Point for DIGIT 1 will be used to display the signed of the result. On represents negative, while off represents positive.

Assignment

1. You have to provide control over the multiplication process in the following way: a user starts a multiplication process by pressing and releasing FLEX_PB1 push button. Before pressing the FLEX_PB1 button user prepares inputs by setting corresponding switches. Any change on switches after the button is released does not result in a change on the LED. If we want new multiplication we should press/release button PB1 again.
2. Write VHDL code for the Booth s algorithm multiplication. (**No credit for non- Booth s algorithm implementation**)
3. Simulate the VHDL design in 2 using ModelSim Simulator for the following test cases:

   0110 x 0011
   1000 x 0010
   0110 x 1110
   1100 x 1001

4. Develop the other necessary components for this designs using schematic capture and/or VHDL.
5. Simulation and design result must be demonstrated to the lab instructor and fully documented in lab report.

Due Day

Lab: 4/16 and Lab Report: 4/23